

10/23/00



jc946 U.S. PTO

10-24-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): JEFFREY WILLIAM JOSTEN
Docket: STL000038US1
Title: EFFICIENT SEQUENCE NUMBER GENERATION IN A MULTI-SYSTEM
DATA-SHARING ENVIRONMENT

jc915 U.S. PTO
09/694185

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

'Express Mail' mailing label number: EL540750244US

Date of Deposit: OCTOBER 23, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

By: Isabell Ogata
Name: Isabell Ogata

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

We are transmitting herewith the attached:

- ☒ Continuation-in-Part Patent Application: Spec. 14 pgs; 41 claims; Abstract 1 pg(s).
The fee has been calculated as shown below in the "Claims as Filed" table.
- ☒ 4 sheets of Drawings
- ☒ A signed Declaration and Power of Attorney
- ☒ Assignment of the invention to INTERNATIONAL BUSINESS MACHINES CORPORATION, Recordation Form Cover Sheet
- ☒ A charge to Deposit Account No. 09-0460 in the amount of \$1,168.00 to cover the Filing Fee
- ☒ A charge to Deposit Account No. 09-0460 in the amount of \$40.00 to cover the Assignment Recordation Fee.
- ☒ Return postcard

CLAIMS AS FILED

Number of Claims Filed	In Excess of:	Number Extra	Rate	Fee
Basic Filing Fee				\$710.00
Total Claims				
41	20	21	x \$18.00	= \$378.00
Independent Claims				
4	3	1	x \$80.00	= \$80.00
MULTIPLE DEPENDENT CLAIM FEE				\$0.00
TOTAL FILING FEE				\$1,168.00

Please charge all fees to Deposit Account No. 09-0460 of I.B.M. Corporation, the assignee of the present application. A duplicate of this sheet is enclosed.

GATES & COOPER
Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, CA 90045
(310) 641-8797

By: George H. Gates
Name: George H. Gates
Reg. No.: 33,500
Initials: GHG/io

EFFICIENT SEQUENCE NUMBER GENERATION
IN A MULTI-SYSTEM DATA-SHARING ENVIRONMENT

CROSS-REFERENCE TO RELATED APPLICATION

5 This application is a continuation-in-part of the following co-pending and
commonly-assigned patent application:

Application Serial No. 09/330,865, entitled "ASSIGNING RECOVERABLE
UNIQUE SEQUENCE NUMBERS IN A TRANSACTION PROCESSING SYSTEM",
filed June 11, 1999, by Jeffrey W. Josten, Chandrasekaran Mohan, and Inderpal S. Narang,
10 attorney's docket number ST9-99-017,
which application is incorporated by reference herein.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

15 The present invention generally relates to computer-implemented transaction
processing systems, and in particular, to a method for efficient sequence number generation
in a multi-system data-sharing environment.

2. Description of Related Art.

20 A database management system (DBMS) usually assigns a unique sequence number
(SN) to fields, records, etc. Generally, the SNs comprise values assigned a monotonically
increasing value in an ascending sequence, although they can encompass other values and

sequences as well.

A problem arises, however, in that the sequence number assignment is an update operation to a record which is locked until the assignment completes. This serializes other applications that also use the sequence number assignment, because they wait for the updated record to be unlocked in order to receive their sequence number assignment. In a multi-system DBMS environment, e.g., where there is data sharing, an update of this record causes serialization across all systems, which inhibits throughput. Furthermore, if a system were to fail while holding the lock on the record, other systems are prevented from accessing the record until restart recovery is performed for the failed system, which inhibits availability.

Thus, there is a need in the art for improved techniques for assigning sequence numbers without serialization.

SUMMARY OF THE INVENTION

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, the present invention discloses a method, apparatus, article of manufacture, and data structure for use in efficiently generating sequence numbers in a multi-system data-sharing environment.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent

corresponding parts throughout:

FIG. 1 schematically illustrates the environment of the preferred embodiment of the present invention;

FIG. 2 illustrates a control page used in the preferred embodiment of the present invention;

FIG. 3 illustrates an in-memory data structure used in the preferred embodiment of the present invention;

FIG. 4 is a flowchart that illustrates the logic performed in assigning sequence numbers from the data structure according to the preferred embodiment of the present invention; and

FIG. 5 is a flowchart that illustrates the logic performed in accessing a next range of sequence number values according to the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following description of the preferred embodiment, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration a specific embodiment in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

Overview

The present invention discloses a method that efficiently assigns recoverable, unique, monotonically-increasing sequence numbers in a multi-system data-sharing environment. Such sequence numbers are often used for database management systems (DBMS's), and
5 other applications.

The sequence numbers in the present invention have no restrictions on their size, or their values. Moreover, a failure of any system does not affect the ability of the other surviving systems to continue generating unique sequence numbers, thereby supporting high availability.

Hardware Environment

FIG. 1 schematically illustrates the environment of the preferred embodiment of the present invention, and more particularly, illustrates a typical distributed computer system 100 using a network 102 to connect one or more clients 104 to multiple servers 106 coupled to one
10 or more direct access storage devices (DASDs) 108. The network 102 may comprise
15 networks such as LANs, WANs, SNA networks, and the Internet. A typical combination of resources may include clients 104 that are implemented on personal computers or workstations, and servers 106 that are implemented on personal computers, workstations, minicomputers, or mainframes.

20 In a preferred embodiment, each of the servers 106 execute a Database Management System (DBMS) 110, which may access shared and non-shared databases 112 stored on the DASDs 108. Moreover, in the present invention, the DBMS 110 controls access to one or

more control pages 114 stored in a catalog within a shared database 112, wherein each of the control pages 114 is used to control the generation and assignment of a recoverable, unique, monotonically-increasing sequence number. All or portions of the control pages 114 may be stored in a data structure 116 in the memory of the servers 106 in order that each instance of

5 the DBMS 110 may access the information stored therein.

Generally, the DBMS 110, databases 112, control pages 114, and data structures 116 are embodied in and/or readable from devices, carriers, media, or signals, such as a memories, data storage devices, and/or remote devices coupled to the computer via data communications devices. Thus, the present invention may be implemented as a method, apparatus, or article of

10 manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term “article of manufacture” (or alternatively, “computer program carrier”) as used herein is intended to encompass any device, carrier, or media that provides access to instructions and/or data useful in performing the same or similar functionality.

15 Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope of the present invention. In addition, those skilled in the art will recognize that any combination of the above components, or any number of different components, including different computers, computer programs, peripherals, and other devices, may be used to implement the present invention, so long as

20 similar functions are performed thereby.

The Control Page

FIG. 2 illustrates the control page 114 that is used in the preferred embodiment of the present invention. The control page 114 stores the sequence number and related attributes, including an identifier 200, a sequence number (SN) 202, a range value (N) 204, a starting sequence number (Starting SN) 206, as well as optional attributes 208.

In the preferred embodiment, the identifier 200 is a user-defined value that identifies the use for the SN 202. Any number of different identifier 200 values could be used, without departing from the scope of the present invention. The identifier 200 is usually (but not always) a string value.

In the preferred embodiment, the SN 202 is the last number that could possibly have been assigned by any instance of the DBMS 110. The SN 202 may be comprised of any number of bits (or bytes or words), so that the SN 202 values are not limited in any way. Generally, the values are assigned in ascending sequence, although any number of different values could be used without departing from the scope of the present invention.

In the preferred embodiment, the range value stored in N 204 is the number of reserved SN 202 assignments that can be made by each instance of the DBMS 110 without accessing the control page 114. N 204 can be initially derived internally by the DBMS 110, or it can be derived from a user-specified value.

In the preferred embodiment, the value stored in Starting SN 206 is the starting value for SN 202 assignments. Starting SN 206 can be initially derived internally by the DBMS 110, or it can be derived from a user-specified value.

Finally, the control page 114 may include any number of optional attributes 208. Moreover, these optional attributes 208 may comprise any number of different data types.

The Data Structure

5 FIG. 3 illustrates the data structure 116 that is used in the preferred embodiment of the present invention. The data structure 116 stores the sequence number and related attributes, including an identifier 300, a sequence number value (SN_MEM) 302, a “number remaining” value (N_REM) 304, as well as optional attributes 306.

10 In the preferred embodiment, the identifier 300 is a user-defined value that identifies the use for the SN_MEM 302 and is matched to a corresponding identifier 200 in the control page 114. Any number of different identifier 300 values could be used, without departing from the scope of the present invention. The identifier 300 is usually (but not always) a string value.

15 In the preferred embodiment, the SN_MEM 302 is the last assigned SN from this instance of the DBMS 110. The SN_MEM 302 may be comprised of any number of bits (or bytes or words), so that the SN_MEM 302 values are not limited in any way. Generally, the values are assigned in ascending sequence, although any number of different values could be used without departing from the scope of the present invention.

20 In the preferred embodiment, the N_REM 304 is the number of reserved SN_MEM 302 assignments that can be made by this instance of the DBMS 110 without accessing the control page 114. N_REM 304 is initially derived from N 204.

Finally, the data structure 116 may include any number of optional attributes 306. Moreover, these optional attributes 306 may comprise any number of different data types.

Assignment of the Sequence Numbers

5 In the preferred embodiment, a control page 114 is stored in a catalog of the database 112 for each SN 202 that may be defined for one or more applications. For example, there might be an SN 202 that is defined for an “order number” field, another SN 202 for a “part number” field, yet another SN 202 for an “invoice number” field, and so on.

10 Preferably, the DBMS 110 includes a Data Definition Language (DDL) that allows the user to define the SNs 202, and to specify the attributes of the control page 114 for each SN 202 (e.g., the identifier 200, current SN 202, range value N 204, starting SN 206, and optional attributes 208). When the SN 202 is first defined, the DBMS 110 initializes the control page 114, and stores the Starting SN 206 value on the control page 114.

15 During operations, the control page 114 is retrieved from the database 112, the information from the control page 114 is stored in the data structure 116 in the memory of each of the servers 106, and the control page 114 is updated. Periodically, the control page 114 may be checkpointed, i.e., saved, to the database 112 on the DASD 108, in order to effect a “hardening” of the control page 114. Generally, this checkpointing is performed by the DBMS 110, in order to provide a protected environment for the control page 114.

20 Redo log records are written as each server 106 updates the SN value in the control page 114. These log records are used for media recovery or restart recovery to reconstruct the SN 202 value in case of failures.

The logic for assigning the next SN involves latching the SN_MEM 302 from the data structure 116, and then using a Compare Double and Swap (CDS) or Compare and Swap (CS) instruction (or similar logic) to atomically read and increment the SN_MEM 302. Similarly, a Compare Double and Swap (CDS) or Compare and Swap (CS) instruction (or
5 similar logic) is used to atomically read and decrement the N_REM 304. After the SN_MEM 302 has been assigned N 204 times, i.e., N_REM 304 reaches zero, then the DBMS 110 instance must access and update the control page 114 to reserve the next range of SN values, wherein the range is indicated by N 204.

Note that in a multi-system data-sharing environment, multiple DBMS 110 instances
10 reserve a range of SN values from the same control page 114, and the use of ranges provides the necessary control over the assignment of SN values. Once a range of SN values has been reserved to the DBMS 110 instance, the SN 202 of the control page 114 is updated to reflect the starting point for the next of SN values. For strict ordering of SN 202 assignments across servers 106, a value of 1 for N 204 can be used.

15 In the preferred embodiment, a P-lock (physical lock) is used to control this access and update to the control page 114 across servers 106. The P-lock is not a "modify" lock, which means that if a server 106 fails while holding the P-lock, the P-lock will not be retained, and thus other servers 106 will not be prevented from continuing to generate new SNs 202, even when the server 106 that is currently holding the P-lock fails.

20 Because the P-lock is non-modify, the control page 114 must be written to external storage (e.g., a coupling facility or DASD 108) and the local copies on the control page 114 in other servers 106 must be invalidated before the server 106 making the update starts

assigning any new SN 202 values. Using a Write Ahead Logging (WAL) protocol, a redo log record is forced to a log file before the control page 114 is written. Alternatively, if the P-lock is made to be a modify lock, the control page 114 does not need to be immediately written.

5 With regard to the checkpointing, the control page 114 may be updated in the database 112 on the DASD 108 after each access and update of the control page 114 by an instance of the DBMS 110 to get the next range of SN values indicated by SN 202 and N 204. Of course, alternative embodiments could update the control page 114 in the database 112 at other intervals as well.

10 The following example further describes the assignment of SN values according to the preferred embodiment of the present invention. In this example, first and second DBMS 110 instances assign SN values from the same control page having an identifier "S1", an SN of 1, an N of 20, and a Starting SN of 1. The first DBMS 110 instance reserves a range of SN values from 1-20, and the second DBMS 110 instance reserves a range of SN
15 values from 21-40. Both the first and second DBMS 110 instances maintain their own data structures 116, including the SN_MEM 302 to indicate the next SN value to assign and N_REM 304 to identify when the range of reserved SN values has been exhausted. As each DBMS 110 instance exhausts its range of reserved SN values, when N_REM 304 reaches
20 zero, the control page 114 is accessed and updated (under the control of a P-lock) to obtain the next range of SN values, e.g., the first DBMS 110 instance might exhaust its range first, and it would access and update the control page 114 to reserve the next range of 20 SN

values, i.e., where the SN values range from 41-60. The P-lock enforces that only one DBMS 110 instance can update the control page 114 at a time.

Note that, while the first DBMS 110 instance is accessing and updating the control page 114, it cannot assign new SN values. However, the second DBMS 110 instance can
5 continue to assign SN values from its data structure 116. Of course, if the second DBMS 110 instance also exhausts its range of SN values, then it also must access and update the control page 114 to reserve the next range of 20 SN values. The P-lock ensures that the updates performed by the first and second DBMS 110 instances to the control page 114 are properly serialized.

Logic for Assigning Sequence Numbers

The following flowcharts describe the processing and logic of initializing the control page 114, assigning the SNs 202, and hardening the control pages 114. This logic is referred to as the NUMA (NUMber Assignment) logic.

Sequence Number Assignment

FIG. 4 is a flowchart that illustrates the logic performed in assigning SN_MEM 302 from the data structure 116 according to the preferred embodiment of the present invention.

Block 400 represents the DBMS 110 latching SN_MEM 302.

Block 402 is a decision block that represents the DBMS 110 determining whether N_REM 304 is greater than 0. If so, control transfers to Block 404; otherwise, control transfers to Block 410.

Block 404 represents the DBMS 110 decrementing N_REM 304.

Block 406 represents the DBMS 110 incrementing SN_MEM 302.

Block 408 represents the DBMS 110 unlatching the SN_MEM 302. Thereafter, the logic terminates.

5 Block 410 represents the DBMS 110 unlatching the SN_MEM 302.

Block 412 represents the DBMS 110 retrieving the next range of "N" sequence number values from the control page 114, as described in FIG. 5. Thereafter, control transfers to Block 400.

10 Retrieve Next Range of Sequence Numbers

FIG. 5 is a flowchart that illustrates the logic performed when retrieving the next range of "N" sequence number values from the control page 114 according to the preferred embodiment of the present invention.

Block 500 represents the DBMS 110 latching the control page 114.

15 Block 502 is a decision block that represents the DBMS 110 determining whether the control page 114 is not yet updated. If so, control transfers to Block 504; otherwise, control transfers to Block 520.

Block 504 represents the DBMS 110 P-locking (non-modify) the control page 114.

20 Block 506 represents the DBMS 110 refreshing the control page 114, if the local buffer storing the control page 114 has been invalidated due to the control page 114 on the DASD 108 being updated from another system.

Block 508 represents the DBMS 110 setting SN_MEM 302 equal to SN 202.

Block 510 represents the DBMS 110 adding the amount N 204 to SN 202.

Block 512 represents the DBMS 110 writing a redo log record into a transaction log, wherein the redo log record provides a restart capability for the sequence number assignment logic.

5 Block 514 represents the DBMS 110 writing the control page 114 to the DASD 108, and then invalidating any copies of the control page 114 stored on other servers 106. This can be done, for example, in an IBM S/390 with one "write-data" request to the coupling facility between servers 106.

Block 516 represents the DBMS 110 setting N_REM 304 to N 204.

10 Block 518 represents the DBMS 110 releasing the P-lock on the control page 114.

Block 520 represents the DBMS 110 unlatching the control page 114.

Thereafter, the logic terminates.

Conclusion

15 This concludes the description of the preferred embodiment of the invention. One of the advantages of the present invention is that sequence numbers can be easily shared among multiple systems, unlike prior art systems that handle only a single system. In addition, a large number of different sequence numbers can be assigned, simply by defining multiple control pages. Moreover, the present invention deals well with failure recovery
20 issues, in that the sequence numbers can be recovered from the control page in the database after a crash. Also, the size of the sequence number is not limited, as in prior art systems.

In summary, the present invention comprises a method, apparatus, article of

manufacture, and data structure for use in efficiently generating sequence numbers in a multi-system data-sharing environment.

The following describes some alternative ways of accomplishing the present invention. Those skilled in the art will recognize that the present invention could be used in any type of computer system. Those skilled in the art also will recognize that different operating environments, transaction processing systems, database management systems, applications, etc., could be substituted for the systems described herein. In addition, those skilled in the art will recognize that the present invention could be used with many types of applications, and need not be limited to the example database management systems described herein.

The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

WHAT IS CLAIMED IS:

1 1. A computer-implemented system for assigning sequence numbers,
2 comprising:
3 (a) a computer system; and
4 (b) sequence number assignment logic, performed by the computer system, for
5 generating a recoverable, unique sequence number for assignment to an application, wherein
6 the sequence number is contained in a control page stored in a database on a data storage
7 device coupled to the computer system and shared with other computer systems.

1 2. The system of claim 1, wherein the control page includes one or more
2 attributes selected from a group of attributes comprising an identifier, the sequence number
3 (SN), a range value (N), and a starting sequence number (Starting SN).

1 3. The system of claim 2, wherein the identifier is a user-defined value that
2 identifies a use for the sequence number.

1 4. The system of claim 2, wherein a value stored in N identifies a range of
2 sequence number assignments.

1 5. The system of claim 1, wherein one or more attributes from the control page
2 are stored in a data structure that is stored in the computer system.

1 6. The system of claim 5, wherein the data structure includes one or more
2 attributes selected from a group of attributes comprising a sequence number (SN_MEM)
3 and a “number remaining” value (N_REM).

1 7. The system of claim 6, wherein a value stored in N_REM indicates when a
2 range of sequence numbers should be obtained from the control page.

1 8. The system of claim 1, wherein the control page is periodically saved to the
2 data storage device, in order to effect a hardening of the control page.

1 9. The system of claim 1, wherein the sequence number assignment logic
2 further comprises logic for latching the sequence number to serialize generation of the
3 sequence number within the computer system.

1 10. The system of claim 1, wherein the sequence number assignment logic
2 further comprises logic for physically locking the control page to serialize updates to the
3 control page across multiple computer systems.

1 11. The system of claim 10, wherein the physical lock is not maintained when the
2 computer system fails, so that other computer systems are not prevented from continuing to
3 generate new sequence numbers.

1 12. A method of assigning sequence numbers in a computer-implemented
2 system, comprising:
3 (a) generating a recoverable, unique sequence number using sequence number
4 assignment logic performed by a computer system for assignment to an application; and
5 (b) storing the sequence number in a control page, wherein the control page is stored
6 in a database on a data storage device coupled to the computer system and shared with other
7 computer systems.

1 13. The method of claim 12, wherein the control page includes one or more
2 attributes selected from a group of attributes comprising an identifier, the sequence number
3 (SN), a range value (N), and a starting sequence number (Starting SN).

1 14. The method of claim 13, wherein the identifier is a user-defined value that
2 identifies a use for the sequence number.

1 15. The method of claim 13, wherein a value stored in N identifies a range of
2 sequence number assignments.

1 16. The method of claim 12, wherein one or more attributes from the control
2 page are stored in a data structure that is stored in the computer system.

1 17. The method of claim 16, wherein the data structure includes one or more

attributes selected from a group of attributes comprising a sequence number (SN_MEM)
and a “number remaining” value (N_REM).

18. The method of claim 17, wherein a value stored in N_REM indicates when a
range of sequence numbers should be obtained from the control page.

19. The method of claim 12, wherein the control page is periodically saved to the
data storage device, in order to effect a hardening of the control page.

20. The method of claim 12, wherein the sequence number assignment step
further comprises latching the sequence number to serialize generation of the sequence
number within the computer system.

21. The method of claim 12, wherein the sequence number assignment step
further comprises physically locking the control page to serialize updates to the control page
across multiple computer systems.

22. The method of claim 21, wherein the physical lock is not maintained when
the computer system fails, so that other computer systems are not prevented from
continuing to generate new sequence numbers.

23. An article of manufacture embodying logic for performing a method of

2 assigning sequence numbers in a computer-implemented system, the method comprising:
3 (a) generating a recoverable, unique sequence number using sequence number
4 assignment logic performed by a computer system for assignment to an application; and
5 (b) storing the sequence number in a control page, wherein the control page is stored
6 in a database on a data storage device coupled to the computer system and shared with other
7 computer systems.

1 24. The article of manufacture of claim 23, wherein the control page includes
2 one or more attributes selected from a group of attributes comprising an identifier, the
3 sequence number (SN), a range value (N), and a starting sequence number (Starting SN).

1 25. The article of manufacture of claim 24, wherein the identifier is a user-
2 defined value that identifies a use for the sequence number.

1 26. The article of manufacture of claim 24, wherein a value stored in N identifies
2 a range of sequence number assignments.

1 27. The article of manufacture of claim 23, wherein one or more attributes from
2 the control page are stored in a data structure that is stored in the computer system.

1 28. The article of manufacture of claim 27, wherein the data structure includes
2 one or more attributes selected from a group of attributes comprising a sequence number

3 (SN_MEM) and a “number remaining” value (N_REM).

1 29. The article of manufacture of claim 28, wherein a value stored in N_REM
2 indicates when a range of sequence numbers should be obtained from the control page.

1 30. The article of manufacture of claim 23, wherein the control page is
2 periodically saved to the data storage device, in order to effect a hardening of the control
3 page.

1 31. The article of manufacture of claim 23, wherein the sequence number
2 assignment step further comprises latching the sequence number to serialize generation of
3 the sequence number within the computer system.

1 32. The article of manufacture of claim 23, wherein the sequence number
2 assignment step further comprises physically locking the control page to serialize updates to
3 the control page across multiple computer systems.

1 33. The article of manufacture of claim 32, wherein the physical lock is not
2 maintained when the computer system fails, so that other computer systems are not
3 prevented from continuing to generate new sequence numbers.

1 34. A data structure used by sequence number assignment logic performed by

the computer, the data structures comprising:

a control page that contains a sequence number that has no restrictions on its size, an identifier that is a user-defined value that identifies a use for the sequence number, a range value (N) that identifies a range of sequence number assignments, and a starting sequence number (Starting SN) that comprises an initial value for the sequence number.

35. The data structure of claim 34, wherein one or more attributes from the control page are stored in an in-memory data structure in the computer system.

36. The data structure of claim 35, wherein the in-memory data structure includes one or more attributes selected from a group of attributes comprising a sequence number (SN_MEM) and a "number remaining" value (N_REM).

37. The data structure of claim 36, wherein a value stored in N_REM indicates when a range of sequence numbers should be obtained from the control page.

38. The data structure of claim 35, wherein the control page is periodically saved to a data storage device, in order to effect a hardening of the control page.

39. The data structure of claim 35, wherein the sequence number is latched to serialize generation of the sequence number within the computer system.

1 40. The data structure of claim 35, wherein the control page is physically locked
2 by a computer system to serialize updates to the control page across multiple computer
3 systems.

1 41. The data structure of claim 40, wherein the physical lock is not maintained
2 when the computer system fails, so that other computer systems are not prevented from
3 continuing to generate new sequence numbers.

EFFICIENT SEQUENCE NUMBER GENERATION
IN A MULTI-SYSTEM DATA-SHARING ENVIRONMENT

ABSTRACT OF THE DISCLOSURE

- 5 A method, apparatus, article of manufacture, and data structure for use in efficiently generating sequence numbers in a multi-system data-sharing environment.

"Express Mail" mailing label number ELJ40756244US
Date of Deposit OCTOBER 23 2000
I hereby certify that this paper or fee is being de-
posited with the United States Postal Service "Express
Post Office to Addressee" service under 37 CFR 1.10 and
the date indicated above and is addressed to:
Assistant Commissioner for Patents, Washington, D.C. 20535.
ISABELL OGATA
(printed name)
Isabell Ogata
(signature)

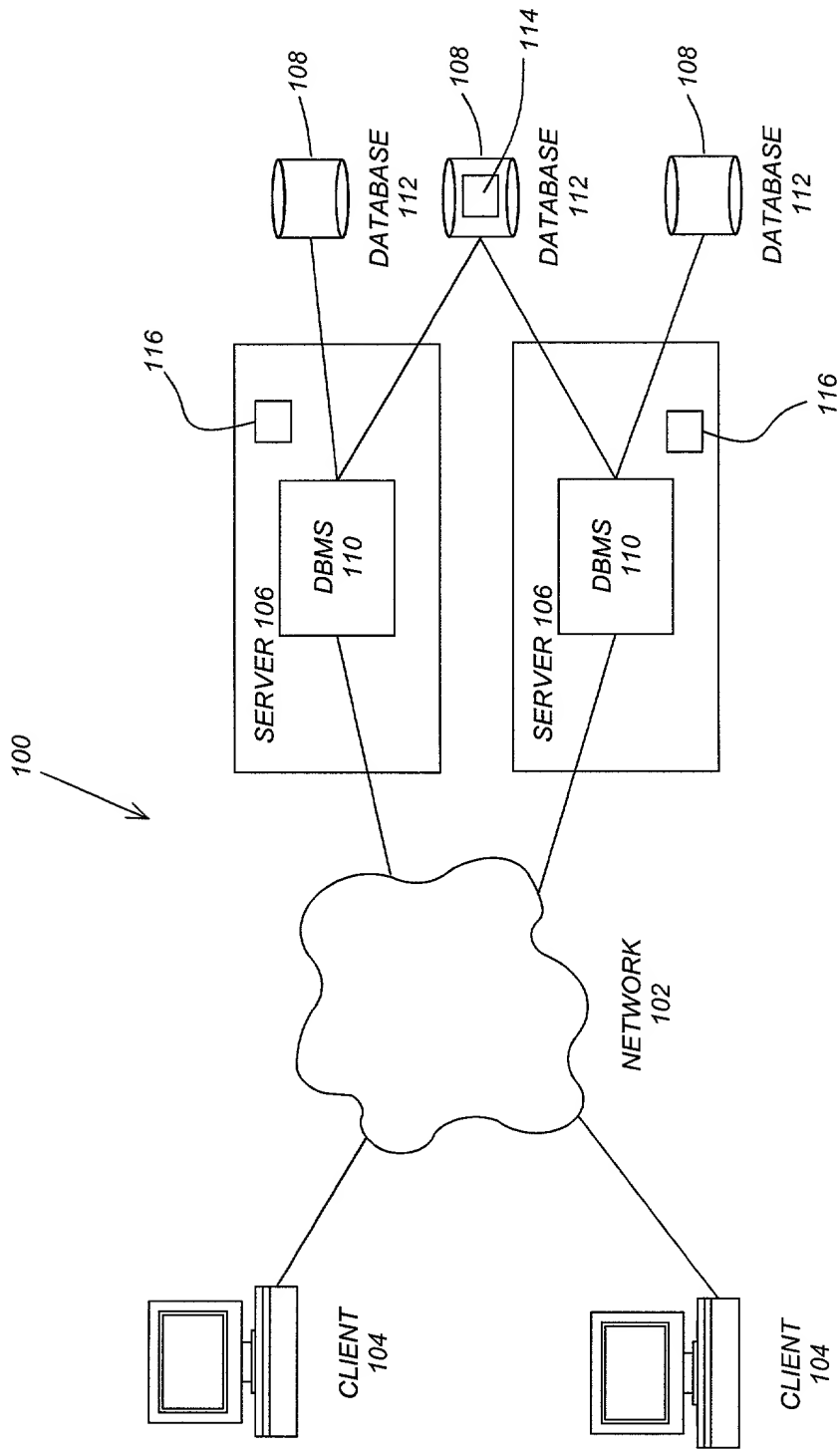


FIG. 1

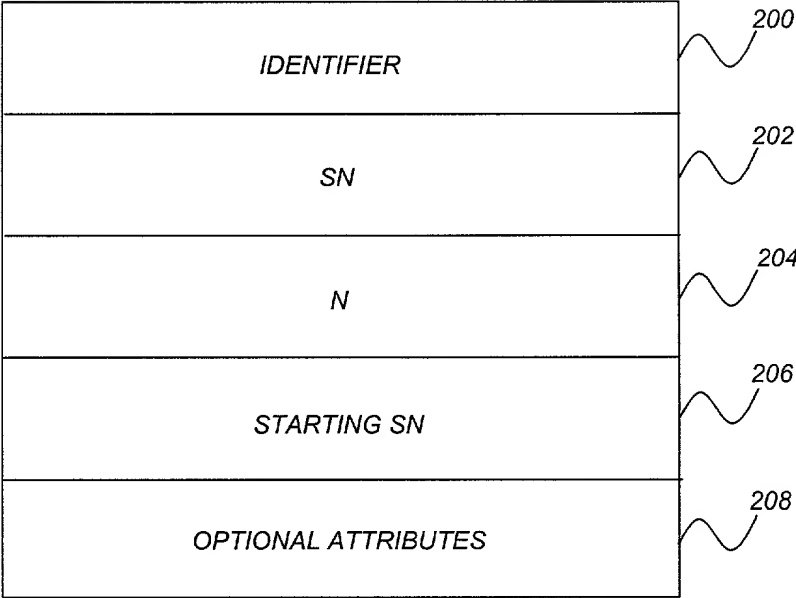


FIG. 2

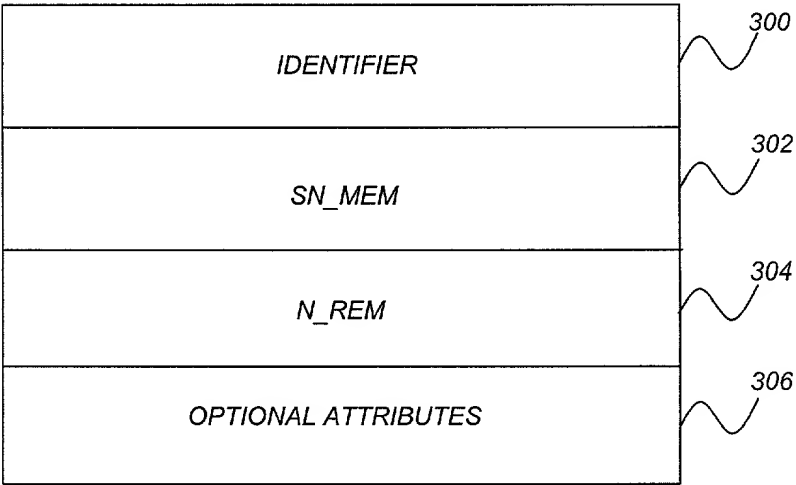


FIG. 3

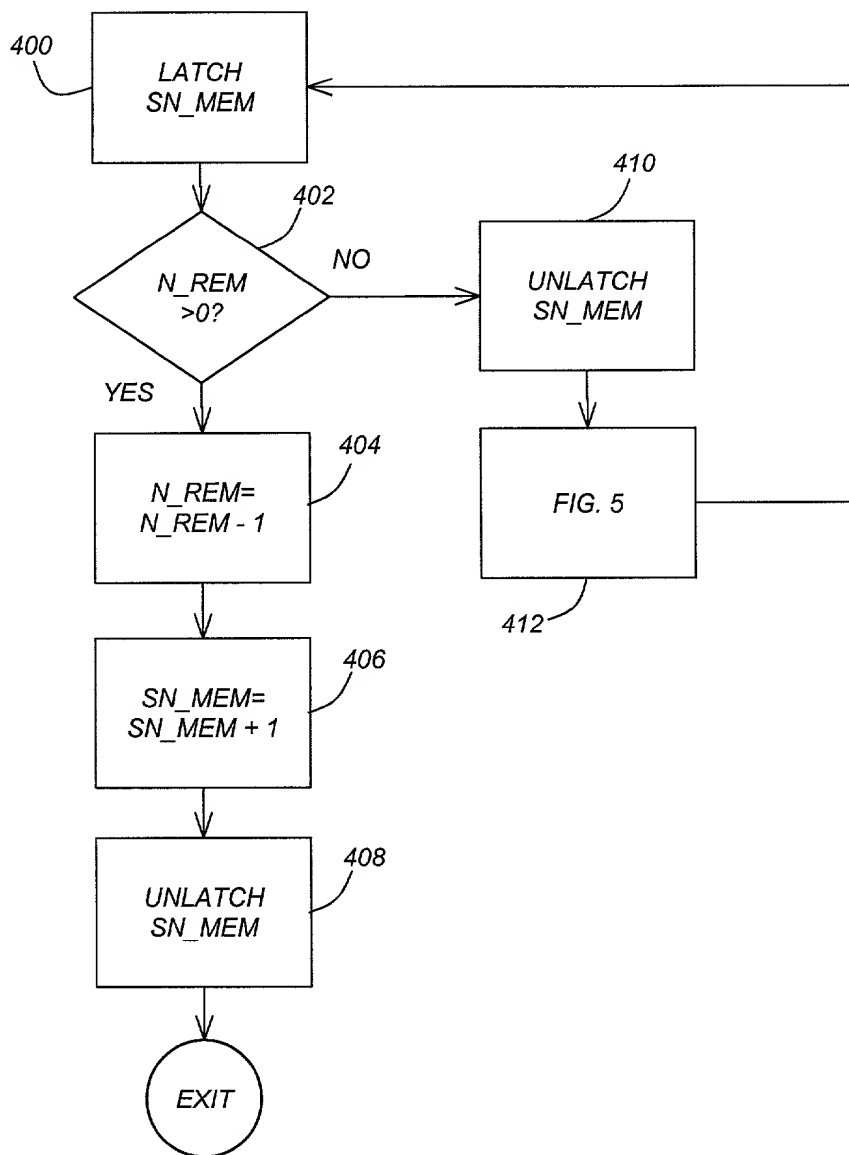


FIG. 4

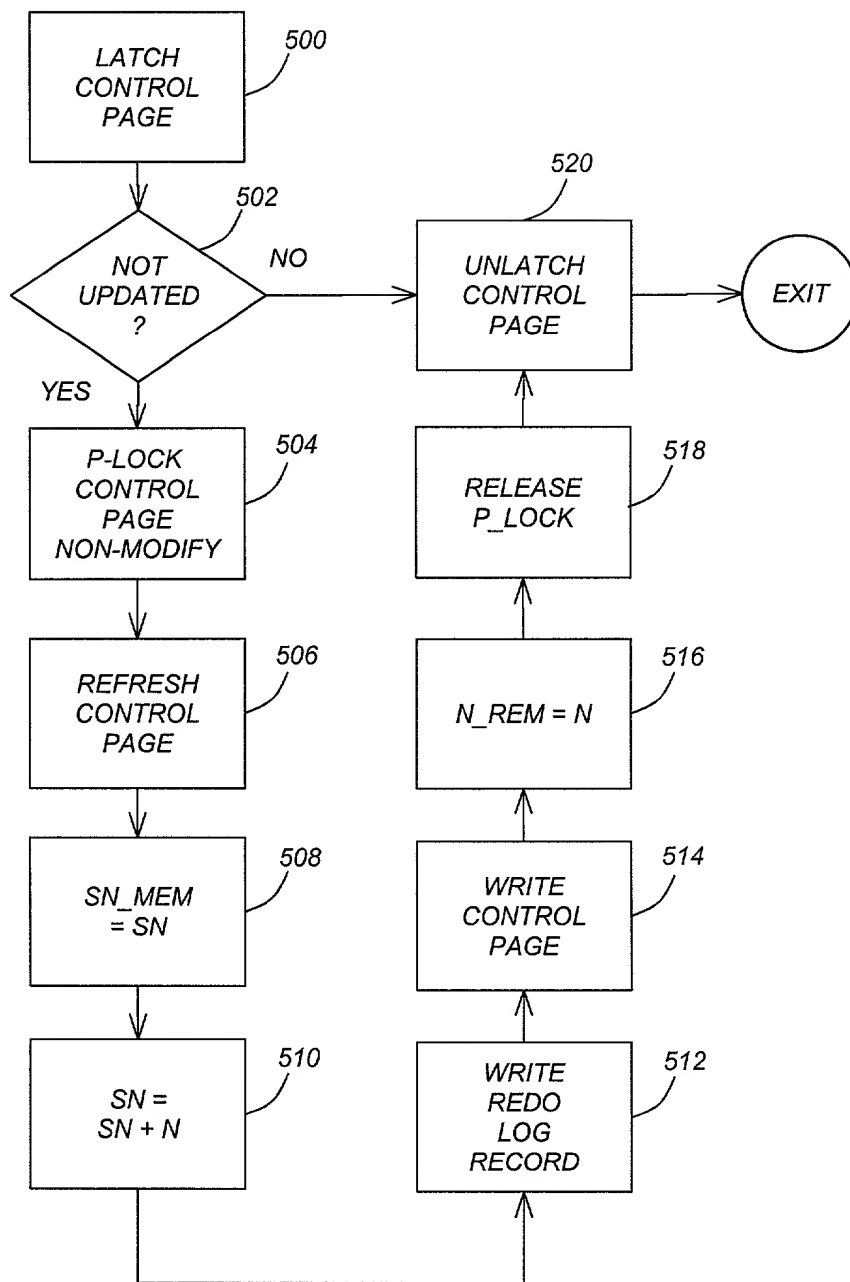


FIG. 5

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL000038US1

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

EFFICIENT SEQUENCE NUMBER GENERATION IN A MULTI-SYSTEM DATA-SHARING ENVIRONMENT

the specification of which (check one)

X is attached hereto.
 _____ was filed on _____
 _____ as Application Serial No. _____
 _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
None			Yes	No
(Number)	(Country)	(Day/Month/Year Filed)		
_____	_____	_____	_____	_____

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

None		
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
_____	_____	_____

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL000038US1

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Timothy M. Farrell, Registration No. 37,321, Ingrid M. Foerster, Registration No. 36,511,
John E. Hoel, Registration No. 26,279, Christopher A. Hughes, Registration No. 26,914,
Prentiss W. Johnson, Registration No. 33,123, Edward A. Pennington, Registration No. 32,588,
Joseph C. Redmond, Jr., Registration No. 18,753, Romualdas Strimaitis, Registration No.
35,697, all of whom are attorneys with IBM Corporation; and

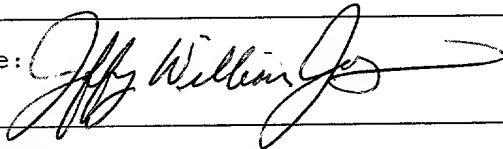
George H. Gates, Registration No. 33,500, Victor G. Cooper, Registration No. 39,641, and
Jason S. Feldmar, Registration No. 39,187, all of whom are attorneys with the law firm of
Gates & Cooper.

Send correspondence to: George H. Gates, Esq.
Gates & Cooper
Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045

Direct Telephone Calls to: (name and telephone number) George H. Gates, (310) 642-4146

Full name of sole or first joint-inventor: Jeffrey William Josten

Inventor's signature:



Date: 10/12/2000

Residence: 6728 Crystal Springs Drive, San Jose, California 95120

Citizenship: United States of America

Post Office Address: Same